Claims

[c1] A Turbo Codes Decoder used as a baseband processor subsystem for iterative decoding a plurality of sequences of received data R_n representative of coded data X_n generated by a Turbo Codes Encoder from a source of original data u_n into decoded data x_n comprising of:

(a) two pipelined SISO Log-MAP Decoders each decoding input data from the other output data in an iterative mode.

(b) the first SISO Log-MAP Decoder having three inputs: R_0, R_1 connecting from ~~the~~ an Input shift register module, and Z_1 feeding-back from ~~the~~ a De-Interleaver Memory module output; and the first Decoder output is connected to an Interleaver Memory module.

(c) the second SISO Log-MAP Decoder having two inputs: R_2 connecting from the Input shift register module, and Z_2 connecting from the Interleaver Memory module output; and the second Decoder output is connected to a De-Interleaver? RAM.

(d) an Interleaver Memory module storing decoded data from the first Log-MAP Decoder, feeding data to the second Log-MA Decoder.

(e) a De-Interleaver Memory module storing decoded data from the second Log-MAP Decoder, feeding-back data to the first Log-MAP Decoder.

(f) an Input Shift Register Buffer storing a block input un-decoded received data, and feeding data to the two Log-MAP Decoders.

(g) a Control logic state machine controlling the overall operations of the Turbo Codes Decoder.

(h) a hard-decoder logic producing a final decision of either logic zero 0 or logic one 1 at the end of the iterations.

[c2] The Decoder system of claim c1, wherein each Log-MAP decoder uses the logarithm maximum a posteriori probability algorithm.

[c3] The Decoder system of claim c1, wherein each Log-MAP decoder uses the soft-input and soft-output (SISO) logarithm maximum a posteriori probability algorithm.

[c4]

The Decoder system of claim c1, wherein the Interleaver Memory module uses a permuter to generate the write-address sequences of the Memory core in write-mode. In read-mode, the memory core read-address are normal sequences.

[c5] The Decoder system of claim c1, wherein the Interleaver Memory module uses dual-port memory RAM.

[c6] The Decoder system of claim c1, wherein the De-Interleaver Memory module uses an inverse-permuter to generate the write-address sequences of the Memory core in write-mode. In read-mode, the memory core read-address are normal sequences.

[c7] The Decoder system of claim c1, wherein the De-Interleaver Memory module uses dual-port memory RAM.

[c8] A method for iterative decoding a plurality of sequences of received data R_n representative of coded data X_n generated by a Turbo Codes Encoder from a source of original data u_n into decoded data x_n comprising the steps of:

(a) coupling two pipelined decoders, having Interleaver Memory and De-Interleaver Memory for storing decoded output and providing feedback input for the decoders.

(b) applying feedback signal from the output of the De-Interleaver Memory to the first decoder by adding the instrinsic values with the received signal input to generate a first decoded output.

(c) applying the first decoded output to the Interleaver Memory using the permuter to generate a memory address for storing the decoded data.

(d) applying the output of the Interleaver Memory to the second decoder with the received signal input to generate a second decoded output

applying the second decoded output to the De-Interleaver Memory using the inverse-permuter to generate a memory address for storing the decoded data

[c9]

An 8-state SISO Log-MAP Decoder for decoding a plurality of sequences of soft-input data SD_0 and SD_1 generated by a receiver to produce decoded soft-output data Y comprising of:

(a) a Branch Metric module computing the two soft-input data SD_0 and SD_1 into 16 branch metric values for each branch in the trellis.

(b) a Branch Metric Memory module storing the 16 branch metric values for each stage k = 0 ...N.

(c) a State Metric module computing state metric values for each state in the trellis using branch metric values. A State Metric Computing module calculates the probability A(k) of each state transition in forward recursion and the probability B(k) in backward recursion.

(d) a State Metric Memory module storing 8 state metric values for each stage k = 0...N.

(e) a Log-MAP module computing the soft decision output based on the branch metric values and state metric values using log maximum a posteriori probability algorithm.

(f) a Control Logic state machine module controlling the overall operations of the Log-MAP decoder.

[c10] The Decoder system of claim c7, wherein the decoder uses the logarithm maximum a posteriori probability algorithm.

The Decoder system of claim c7, wherein the decoder uses an implementation of state-metric in forward recursion with Add-Compare-Select (ACS).

The Decoder system of claim c7, wherein the decoder uses an implementation of state-metric in backward recursion with Add-Compare-Select (ACS).

[c11] The Decoder system of claim c7, wherein the decoder uses the soft-input and soft-output (SISO) logarithm maximum a posteriori probability algorithm.

[c12] The Decoder system of claim c7, wherein the decoder uses a convolutional 8-states trellis state transition diagram.

[c13] The Decoder system of claim c7, wherein the the branch metric module uses a binary adder, a binary Subtracter, and two binary two-complementers logic.

[c14] The Decoder system of claim c7, wherein the the state metric module uses a binary adder, a comparator, a Mux selector logic.

[c15] The Decoder system of claim c7, wherein the the log-map module uses binary adders, binary maximum selectors logic.

[c16] The Decoder system of claim c7, wherein the soft decoder module uses soft decision algorithm.

[c17] The Decoder system of claim c7, wherein the the branch metric memory module uses dual-port memory RAM.